

Provisioning a Restful Pedagogical Service in the Cloud

Shueh-Cheng Hu^{#1}, I-Ching Chen^{#2}, Yaw-Ling Lin^{#3}

^{#1}*Dept. of Computer Sci. & Communication Engineering*

Providence University, Taichung, Taiwan, ROC

¹*schu@pu.edu.tw*

^{#2}*Dept. of Information Management*

Chung Chou University of Science and Technology, Changhua, Taiwan, ROC

²*jine@dragon.ccut.edu.tw*

^{#3}*Dept. of Computer Science & Information Engineering*

Providence University, Taichung, Taiwan, ROC

³*yllin@pu.edu.tw*

Abstract— Online education and training play critical roles in the age of Internet. For easy sharing and reusing pedagogical resources that are high-priced to develop, there are standards for encoding instructional contents and regulating their presentational formats. However, because access interfaces and message packing schemes of most e-learning resources are still proprietary, the offered contents are available to privileged users only and thus are difficult to utilize and circulate. SOAP-based Web services to some extent solve the openness issue by enforcing contents and their packing schemes conform to pre-defined formats. To enrich online pedagogical resources and facilitate sharing them among educators and learners, it is of great importance to build them in not only open, but also efficient and easy-accessible way. This article proposes an approach for provisioning pedagogical resources in the cloud based upon the representational state transfer architectural (REST) style. The anticipation is that the constructed services enable instructors and students to easily use a variety of devices to access pedagogical resources under diverse learning circumstances, which is realization of the so-called ubiquitous learning.

Keywords— e-learning; pedagogical resources; cloud computing; SaaS; Restful Web services

I. INTRODUCTION

Many organizations have embraced e-learning technologies and programs to train employees or serve their customers. Comparing with conventional classroom training, the merits brought by e-learning are the flexible time and location for conducting learning activities, as well as adaptive instruction materials. To build qualified e-learning environments, much effort are necessary to maintain online instructional materials up-to-date and satisfactory. Another issue is to meet diverse education needs, instructional packages usually need to be delivered in different contexts and at different level of difficulty. To manage and deliver online pedagogical resources in a flexible and efficient way, this article takes online question bank as an example to clarify the corresponding

concepts and illustrate the design rationales of the proposed solution.

Online question bank systems, or known as e-assessment modules in the e-learning context, acquire and store contents comprising test questions and the corresponding answers in advance, and then deliver requested contents on demand. Despite there are some concern regarding the effectiveness of their usage, they do have advantages such as dynamic composition of test, promptly providing questions and instantly receiving feedback so instructors are able to identify learners' strength and weakness [1] whenever they intend to do so.

To enrich banks' contents, it is necessary to harness collective intelligence and instructional experience from educators. Researchers have been trying to find solutions to facilitate the compatibility among different question banks [2]. The most well-known concrete result is the question & test interoperability (QTI) specification [3], which is currently regulated and maintained by the IMS Global Learning Consortium. However, the QTI only enforces the compatibility among the contents from different question banks by applying a common representational schema in XML. Beyond the contents' compatibility issue, an open and easy-to-access interface to online question banks is vital to facilitate the utilizing, reusing, and sharing of valuable contents from educators with diverse perspectives and instructional experiences.

Among many advantages of service-based pedagogical resources [4], the most attractive one to

both content consumers and providers is the open access method that are based on existing standardized techniques including HTTP, XML, and Simple Object Access Protocol (SOAP), etc. Besides being able to couple distributed and heterogeneous software applications, the standardization and the derived openness and interoperability ease the access to service-based resources and the composition of them. The traditional SOAP-based Web applications apply the SOAP to encapsulate the messages being transferred between service consumers and providers. Consequently, the composing and parsing of SOAP messages result in significant processing overhead not only on server sides, but also on client sides, which usually own inferior processing power than computers with conventional form factors. The overhead not only affects the performance adversely, also indirectly restrains users' selectivity of devices that they can use to efficiently consume available online services [5].

To resolve the aforementioned issues, the representational state transfer (REST) architectural style [6] offers a promising approach. Accordingly, this article proposed an online question bank that is designed with restful Web service and the associated techniques. The proposed architecture embodies merits including lightweight, pervasive access, scalability, and flexibility.

The remaining parts of this paper are organized as follows: Section II briefs related techniques and the prior works. Section III discusses the design rationales of the proposed architecture. Concluding remarks and future works are outlined in the last section.

II. RELATED TECHNIQUES AND PRIOR WORKS

The cloud learning or service-oriented learning approach fundamentally changes the way of provisioning pedagogical resources. Basically, pedagogical resources are prepared in advance and accessible through Web service providers during instructional process, as Fig. 1 illustrates. In other words, all pedagogical resources in the cloud could be wrapped and offered in the form of service, and then, learning management systems can

dynamically request required services on demand and on behalf of learners.

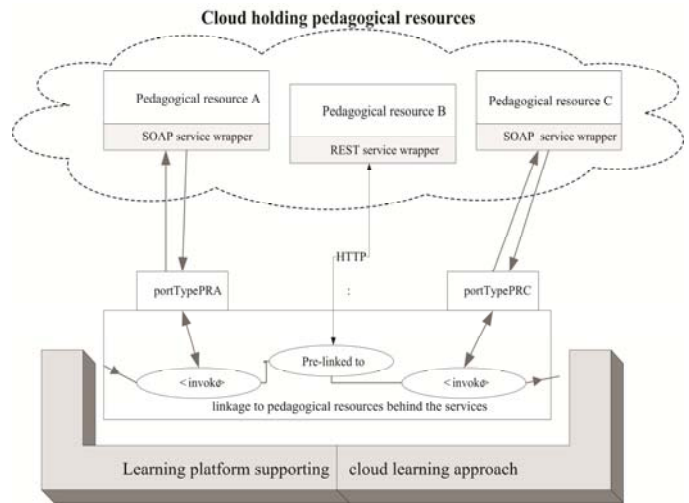


Fig. 1 The provision of learning materials in cloud learning environments

In summary, the service-oriented approach for conducting Web-based instructional processes will bring a number of advantages over the current approach. Besides the general ones such as flexible and evolvable structure, distributed and cross-platform operational environment, those in the context of e-learning include first, the improved recentness and consistency of instructional materials. Because instructional resources access through dynamic linkage ensures that learners can retrieve the latest updated instructional contents from providers on demand. On the other side, there will be reduced inconsistency among different courses that share the same resource, that is because every user acquires the contents dynamically, instead of duplicating and embedding individual copies to different course packages separately. Second, an unleashed, or at least more extensible, scope of reachable pedagogical resources is possible. This aspect allows learners to experience best available learning contents, no matter where the contents are located. In other words, all pedagogical resources through accessible services rather than only those pre-packed in a collection could serve learner's needs upon request. This definitely will enrich the learning experiences of all types of learners. Third, from the perspective of e-learning contents providers, the service-oriented approach of delivering instructional contents enables the centralized control scheme over

learning resources. In other words, a centralized-paradigm for managing learning resources allows the providers of e-learning contents to conduct access control, right management, and maintenance works more efficiently and effectively, comparing with the monolithic approach.

When Web service techniques were applied in the provision of e-learning resources, its platform-neutrality characteristic means no matter which implementation techniques and operating platforms were chose, once an instructional resource was developed or just wrapped as a service, all clients on the Internet can send compliant request to access it. Obviously, that will leverage the reusability and sharing of learning objects and instructional processes beyond the current level, which is conducted by the SCORM-compliant e-learning products.

Basically, there are two kinds of Web service enabling technologies: SOAP (Simple Object Access Protocol) based and REST (REpresentational State Transfer) styled. Due to the overhead of volume message processing, the SOAP-based Web services became bulky to sites with intensive traffic. To address the issue, the REST-styled service presented by Roy Fielding in his dissertation [6] looks a promising approach for architecting various Web services that need to serve vast amount of user requests.

The Restful Web applications [7] emerged as an lightweight alternative for realizing the concept of software as a service (SaaS) [8, 9], which is one of the major service delivery models in cloud computing [10] environment, the other two service models are infrastructure as a service (IaaS) and platform as a service (PaaS). In a SaaS model, functionalities are delivered to users through the Internet; users do not need executable files and data, instead, they are hosted in the cloud. Basically, users can consume SaaS via a Web client software (browser) that might run on diverse form factors such as smart phones, thin clients, tablet PCs, desktops, and so on. Besides offering a fairer pricing scheme that charge users based on real usage, the SaaS model accelerated functions and

data delivery, this feature is very critical to modern learning platforms where people keep up with new knowledge and skills for better competitiveness.

In a restful Web application, everything that could be accessed or operated are treated as resources. The resources must be identifiable via an uniform naming scheme, and the uniform resource identifier (URI) is practically used in all restful Web applications. In contrast with its heavyweight counterpart: the SOAP-based Web service, the restful Web service associates standard HTTP methods with operations that intended to be performed on resources. In general, the four basic HTTP methods: PUT, GET, POST, and DELETE have been used to symbolize the create, retrieve, update, and delete operations on resources, respectively. For example, to identify a pedagogical resource comprising a bank of questions on a specific subject: Java programming, the provider can use the URI like <http://www.onlineqb.org/banks/java/>; to retrieve all question items from that resource, a HTTP GET request could be sent to the URI; identifier of the target resource.

Due to the open and uniform identifying scheme and access (operation) interface, the restful approach significantly reduces the complexity that are caused by the required processing of SOAP-based messages; either message composition or decomposition. Consequently, message recipients can receive and then directly interpret (map) the request sent by users. Such reduction offers users and bystanders a lightweight option for utilizing Web services in the cloud. Moreover, the plain HTTP-based access interface facilitates the integration of popular and lightweight techniques such as asynchronous JavaScript and XML (AJAX) into client side of restful Web applications, which further enables more service consumers using devices with various form factors to access online pedagogical resources ubiquitously.

Not surprisingly, the aforementioned strengths motivated flagship online enterprises including Yahoo, Facebook, and Google that must handle millions of user requests daily, to aggressively adopt the restful approach for

developing their Web applications and publish the corresponding APIs with an eye to the performance and pervasiveness. Besides online enterprises, research organizations such as the Lawrence Berkeley National Laboratory also developed a restful API for scientists, thus they can remotely access an array of high performance computing (HPC) resources that were deployed within the laboratory via light-weight mobile devices with Web browser [11]. Besides, an early feasibility prototype using restful Web service in telecommunication industry also indicated that the REST architectural style is also suitable for bridging services across technologies and application domains [12]. In summary, the restful style have being applied to develop Web applications with diverse purposes [13-16].

III. DESIGN RATIONALES OF THE PROPOSED ARCHITECTURE

The proposed architecture of online question bank consists of major components as shown in Fig. 2. Fundamentally, the architecture is similar to the multi-tier structure of typical Web applications. The noticeable features include the usage of the restlet [17], an API and a reference implementation of Java-based restful Web service interface (JAX-RS 1.0). The restlet framework aims to simplify and speed up the development of restful Web applications with Java technologies. The implementation of this work chose restlet Java EE edition, which allows our restful Web services run on regular JVMs or in servlet containers. The Apache Tomcat was used as a servelt container to host resources that were developed within restlet framework.

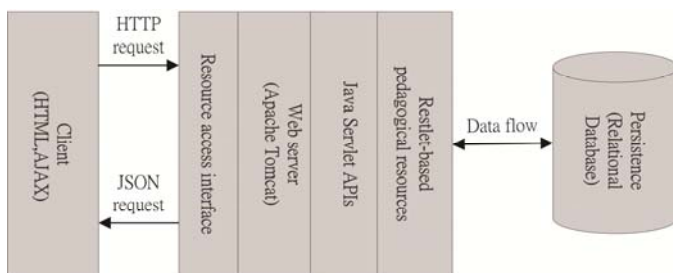


Fig. 2 The architecture of the proposed pedagogical resources provisioning

Regarding the response messages, typical restful applications allows that resources have

different representation formats, e.g. plain text, XML, and JavaScript Object Notation (JSON) [18], etc. The client usually can ask for specific representation via the content negotiation protocol. In the proposed design, JSON is chosen due to its native compatibility with JavaScript-enabled Web clients. On the client side, any devices running browsers that can handle a particular AJAX object: XMLHttpRequest can be served as a platform for using and administering available restful resources.

Another feature is that the uniform interface for accessing all resources in this question bank, which include sub-banks with items in particular subject, question items, and the corresponding answer items that are persisted in a relational database. The design of the interface for accessing this question bank complied with the following constraints: first, all types of resources in the bank will be organized in hierarchical style, e.g., the bank consists several of sub-banks that in turn comprising questions associated with a specific subject domain, and one question may contain a number of valid answers. Second, identifiers of all resources follow a common URI pattern, and thus could be represented hierarchically. For example, to identify a particular (say, the 100th one) question item in a sub-bank focusing on specific subject domain: Java programming, we can use <http://www.onlineqb.org/banks/java/100>, correspondingly, the 2nd valid answer to this question item could be identified via <http://www.onlineqb.org/banks/java/100/2>. Third, the four basic HTTP methods: POST, GET, PUT, and DELETE are uniformly used to map the CRUD (Create, Read, Update, Delete) operations on a resource, respectively. Accordingly, to create a new instance for the 2nd valid answer of the 100th question item, a POST request including the HTTP verb and the answer contents would be sent to <http://www.onlineqb.org/banks/java/100/2>.

In general, the APIs for accessing question items about a particular subject such as Java programming, could be formatted as TABLE I shows. Overall speaking, all the features and constraints mentioned above will make the online

question bank lightweight (high performance), easy to use (pervasiveness), and scalable (elasticity).

TABLE I
API DESCRIPTIONS OF RESTFUL RESOURCES

API function	HTTP method	URL path
Retrieve all items	GET	/banks/java
Create a new item	POST	/banks/java
Retrieve details of a particular question item	GET	/banks/java/[item_id]
Change details of a particular question item	PUT	/banks/java/[item_id]
Delete a particular question item	DELETE	/banks/java/[item_id]

IV. CONCLUSIONS

Cloud computing offers people a new approach to utilize computing resources, either hardware or software. Comparing with the present usage of computing resources, the new approach is more elastic, fairer in terms of pricing scheme, and energy-efficient.

The "software as a service" concept in the cloud computing field, can realize the pay-per-use model, has being substantially reshaped the way that people develop, utilize and maintain software. Basically, there are two techniques for realizing the SaaS concept: SOAP and REST. Comparing with its SOAP-based counterpart, restful Web service is relatively favorable in terms of implementation complexity and message processing overhead. This simplicity makes restful Web applications not only lightweight, but also easier to use from the perspective of patrons who need to use various devices to access resources in the cloud.

This article proposed an restful approach for architecting pedagogical resources such as an online question bank, which aims to facilitate prompt assessment works of learning activities under diverse circumstances. Besides, the restful design can make pedagogical services accommodate more users with the fixed investment of computing devices. The ubiquity of restful pedagogical services will be able to spawn various

education mashups, which are lightweight applications combining data, presentation and functionality from different service providers. These mashups can easily construct a specific learning environment according to instructors and learners' needs, which is very difficult if not impossible to be realized through integrating on-premises software systems.

The design patterns and experience learned in this work also could be applied to develop other online services. Although the feasibility of the proposed approach looks promising based on prior successful works with restful architectural style, the widely acceptance of it needs further works on session management and security. For the sake of scalability, restful services supposed to be completely stateless. In other words, using widely accepted server-side session mechanism violates the constraints of the restful design. However, it is obvious that access control and session management mechanisms are vital to practical pedagogical services.

ACKNOWLEDGMENT

This research work has being funded by the grant from the National Science Council, Taiwan, ROC, under Grant No. NSC 99-2632-E-126-001-MY3.

REFERENCES

- [1] C. Keing, J. Lo, P. Lam, and C. McNaught, "Summative eAssessments: Piloting acceptability, practicality and effectiveness," in *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* Vancouver, Canada: AACE, 2007.
- [2] W. Premchaiswadi, S. Pangma, and N. Premchaiswadi, "Knowledge Sharing for an On-Line Test Bank Construction," in *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* Vancouver, Canada: AACE, 2007.
- [3] I. G. L. Consortium, "IMS Question & Test Interoperability Specification," 2011.
- [4] S.-C. Hu and I. C. Chen, "The service-oriented approach for conducting web-based instructional processes—advantages and challenges," *British Journal of Educational Technology*, vol. 39, pp. 944-947, 2008.
- [5] G. Mulligan and D. Gracani, "A comparison of SOAP and REST implementations of a service based interaction independence middleware framework," in *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 2009, pp. 1423-1432.
- [6] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," in *Information and Computer Science*. vol. DOCTOR OF PHILOSOPHY Irvine, California: UNIVERSITY OF CALIFORNIA, IRVINE, 2000, p. 90.

- [7] S. Vinoski, "RESTful Web Services Development Checklist," *Internet Computing, IEEE*, vol. 12, pp. 96-95, 2008.
- [8] F. Liu, W. Guo, Z. Q. Zhao, and W. Chou, "SaaS Integration for Software Cloud," in *2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 402-409.
- [9] G. Goth, "Software-as-a-Service: The Spark That Will Change Software Engineering?," *Distributed Systems Online, IEEE*, vol. 9, pp. 3-3, 2008.
- [10] T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," in *24th IEEE International Conference on Advanced Information Networking and Applications*: IEEE, 2010, pp. 27-33.
- [11] S. Cholia, D. Skinner, and J. Boverhof, "NEWT: A RESTful service for building High Performance Computing web applications," in *Gateway Computing Environments Workshop (GCE)*, 2010, pp. 1-11.
- [12] C. Fu, F. Belqasmi, and R. Glitho, "RESTful web services for bridging presence service across technologies and domains: an early feasibility prototype," *Communications Magazine, IEEE*, vol. 48, pp. 92-100, 2010.
- [13] P. Belimpasakis and S. Moloney, "A platform for proving family oriented RESTful services hosted at home," *Consumer Electronics, IEEE Transactions on*, vol. 55, pp. 690-698, 2009.
- [14] H.-M. Rissanen, T. Mecklin, and M. Opsenica, "Design and Implementation of a RESTful IMS API," in *The 6th International Conference on Wireless and Mobile Communications (ICWMC)*, 2010, pp. 86-91.
- [15] S. Gao, H. Yu, Y. Gao, and Y. Sun, "A design of RESTful style digital gazetteer service in cloud computing environment," in *The 18th International Conference on Geoinformatics 2010*, pp. 1-6.
- [16] Z. Qian and W. Xianglong, "The research and implementation of a RESTful map mashup service," in *2010 Second International Conference on Communication Systems, Networks and Applications (ICCSNA)*, vol. 1, 2010, pp. 401-403.
- [17] Noelios Technologies, "Restlet - The leading RESTful web framework for Java," <http://www.restlet.org/>, last access: 2011/09/01, 2011.
- [18] Douglas Crockford, "JSON - JavaScript Object Notation," <http://www.json.org/>, last access: 2011/08/21, 2011.